Лабораторная работа №3

Распределение по вариантам:

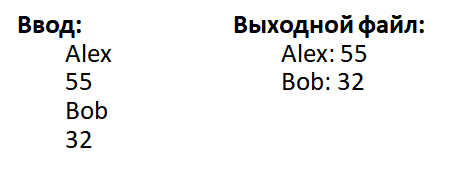
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ варианта** | **Задача 1** | **Задача 2** | **Задача 3** | **Задача 4** |
| **1** | 1 | 1, 5 | 1 | 1 |
| **2** | 2 | 2, 5 | 2 | 2 |
| **3** | 3 | 3, 5 | 3 | 3 |
| **4** | 4 | 4, 5 | 4 | 4 |
| **5** | 5 | 1, 5 | 5 | 5 |
| **6** | 6 | 2, 5 | 6 | 1 |
| **7** | 7 | 3, 5 | 7 | 2 |
| **8** | 8 | 4, 5 | 8 | 3 |
| **9** | 9 | 3, 5 | 9 | 4 |
| **10** | 10 | 4, 5 | 10 | 5 |

# Задача №1. Простая работа со строками, списками, словарями и файлами

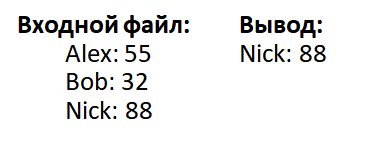
1. Считать числа через пробел, вывести количество различных чисел.
2. Даны два списка чисел. Найдите все числа, которые входят как в первый, так и во второй список и выведите их в порядке возрастания.
3. Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово YES (в отдельной строке), если это число ранее встречалось в последовательности или NO, если не встречалось.
4. Определите, сколько различных слов содержится во введённом пользователем тексте.
5. Выборы. Перечислено кол-во проголосовавших жителей каждого города за различных кандидатов президентов. Необходимо рассчитать кто выиграет голосование (и сколько наберет процентов).
6. Роберт играет в компьютерные игры. На протяжении N часов компьютер каждый час фиксирует информацию о том, играл ли Роберт в компьютер. Необходимо сказать в каком периоде K часов Роберт играл большего всего.
7. Даны номера ударения в словах в списке, затем вводится предложение. Необходимо расставить в нем ударения (заменить строчную букву на заглавную в слове).
8. Дан текст. Выведите все слова, встречающиеся в тексте и укажите сколько раз оно встретилось в тексте.
9. Дан текст в файле. Выведете ТОП-3 самых популярных словосочетаний в тексте.
10. Вам дан словарь, состоящий из пар слов. Каждое слово является синонимом к парному ему слову. Все слова в словаре различны. Для слова из словаря, записанного в последнем предложении текста, определите его синоним.

# Задача №2. Работа с текстовыми файлами

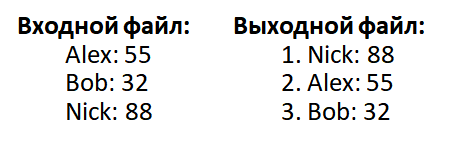
1. Считать текст из файла input.txt и записать в файл output.txt.
2. Создать файл results.txt, куда записать результаты игроков считая имя игрок и его результат с клавиатуры.



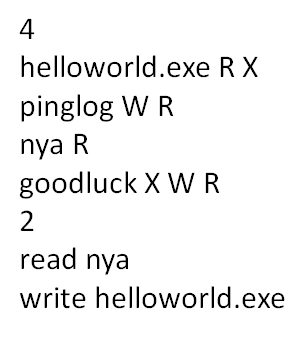
1. Считать результаты игроков из файла results.txt и вывести на экран лучшего игрока по количеству баллов.



1. Считать результаты игроков из файла results.txt и записать в файл output.txt рейтинг игроков по возрастанию их количества баллов.



1. Для каждого файла известно, с какими действиями можно к нему обращаться: запись W, чтение R, запуск X. В первой строке содержится число N — количество файлов содержащихся в данной файловой системе. В следующих N строчках содержатся имена файлов и допустимых с ними операций, разделенные пробелами. Далее указано чиcло M — количество запросов к файлам. В последних M строках указан запрос вида **Операция Файл**. К одному и тому же файлу может быть применено любое количество запросов. Вам требуется восстановить контроль над правами доступа к файлам (ваша программа для каждого запроса должна будет возвращать OK если над файлом выполняется допустимая операция, или же Access denied, если операция недопустима.



# Задача №3. Работа со словарём

Создайте словарь в котором хранятся пятнадцать пар ключ-значение следующего типа (согласно варианту):

1. Фамилии студентов и их рейтинг.
2. Марки автомобилей и их мощность.
3. Фильмы и их рейтинг.
4. Суда и их грузоподъемность.
5. Файлы файловой системы и его размер.
6. Звуковой файл и его продолжительность в секундах.
7. Самолеты и их крейсерская скорость.
8. Города и их население.
9. Аэропорты и их пропускная способность.
10. Страны и их площадь.

Ключи изначально хранятся в list(), словарь должен формироваться из элементов листа, при помощи случайных значения из диапазона приближенного к реаляьному для данного ключа.

При помощи меню реализуйте следующие методы:

1. Задать новые случайные значения словаря.
2. Добавить элемент словаря вручную при помощи ввода пары ключ:значение.
3. Вывести в консоль весь словарь при помощи **pprint.** Пример:

from pprint import pprint

d = {'file1.mp3':251, 'file2.flac':149}

pprint(d, width=1)

1. Вывести ключи с наибольшим, наименьшим значением и общее среднее значение.
2. Ввести ключ элемента, проверить наличие элемента в словаре без применения исключений.
   1. Если введенный ключ существует, удалите его.
   2. Если после удаления словарь пуст, в меню отображается только 1 и 2 пункт.
3. Приведите словарь к виду:

d = [{'key': 'file1.mp3', 'value': 251}, {'key': 'file2.flac', 'value': 149}]

* 1. Полученный список выведите при помощи **pprint.**
  2. Произведите сортировку полученного списка по значениям при помощи лямбда функции (рассматривали в третьей презентации с 13 слайда).
  3. Выведите список при помощи **pprint.**

1. Задать значение элемента по введенному ключу.
2. Вывести на экран ключи с одинаковыми значениями.

# Задача №4. Введение в базы данных

Реализуйте хранение и необходимые методы обработки (добавление / удаление / редактирование данных, вывод данных по параметру) согласно заданной предметной области, пример реализации структуры данных указанный в варианте, может быть изменен.

Хранение желательно реализовать в JSON файле. Формат JSON схож со словарём, файл в формате JSON имеет расширение .json. Можно реализовать через текстовые файлы.

Пример работы с JSON: <https://www.geeksforgeeks.org/reading-and-writing-json-to-a-file-in-python/>

**Вариант 1**

**Библиотека**

{‘Книги’: [{‘Название’: value, ‘Авторы’:[value, value, value], ‘Издание’: value, ‘Год издания’: value,... }], ‘Читатели’ : [{‘ФИО’: value, ‘История’: [{‘id’ : value, ‘Дата получения’: date value, ‘Дата сдачи’: date value}, ...]}]}

Структура данных читателя не должна содержать информации о книгах кроме ID, при выводе истории необходимо выводить данные исходя из связки по ID.

**Вариант 2**

**Автобусный парк**

{‘Автобусы’: [{‘id’: value, ‘Гос. номер’: value, ‘Маршрут’: value, ‘Дата ТО’: value, ...}], ‘Водители’: [{‘ФИО’: value, ‘История’: [{‘id’ : value, ‘Дата выезда’: datetime value, ‘Дата сдачи смены’: datetime value}, ...]}]}

Структура данных водителей не должна содержать информации о автобусах кроме ID, при выводе истории необходимо выводить данные исходя из связки по ID.

**Вариант 3**

**Аэропорт**

{‘Самолеты’: [{‘id’: value, ‘Номер’: value, ‘Модель’:value, ‘Маршрут’: value, ‘Дата ТО’: value, ...}], ‘Пассажиры’: [{‘ФИО’: value, ‘История’: [{‘id’ : value, ‘Дата полета’: date value, ‘Место’: value}, ...]}]}

Структура данных пассажирова не должна содержать информации о самолетах кроме ID, при выводе истории необходимо выводить данные исходя из связки по ID.

**Вариант 4**

**Продуктовый склад**

{‘Продукты’: [{‘id’: value, ‘Название’: value, ‘Условия хранения’:value, ‘Срок годности’: value, ...}], ‘Поставщики’: [{‘Название’: value, ‘История’: [{‘id’ : value, ‘Дата поставки’: date value, ‘Место’: value}, ...]}]}

Структура данных поставщиков не должна содержать информации о продуктах кроме ID, при выводе истории необходимо выводить данные исходя из связки по ID.

**Вариант 5**

**Аренда автомобилей**

{‘Автомобиль’: [{‘id’: value, ‘Гос. номер’: value, ‘Марка’: value, ‘Модель’: value, ‘Дата ТО’: value, ...}], ‘Водители’: [{‘ФИО’: value, ‘История’: [{‘id’ : value, ‘Дата начала аренды’: datetime value, ‘Дата окончания аренды’: datetime value}, ...]}]}

Структура данных водителей не должна содержать информации о автомобилях кроме ID, при выводе истории необходимо выводить данные исходя из связки по ID.